Module 6x – iBGP and Basic eBGP

Objective: Simulate four different interconnected ISP backbones using a combination of IS-IS, internal BGP, and external BGP.

Topology :

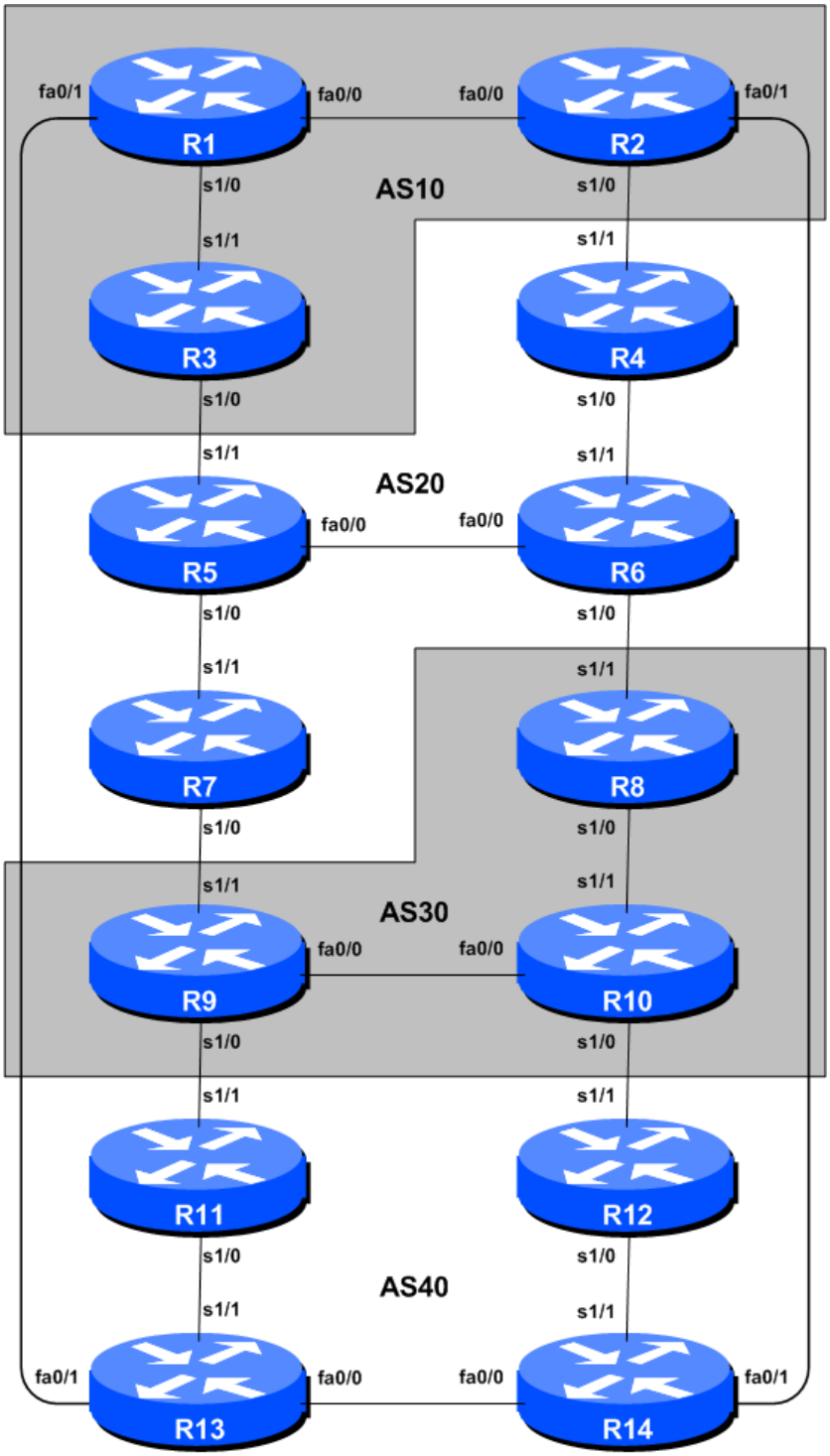

Figure 1 – BGP AS Numbers

Lab Notes

The purpose of this module is to build a basic 4 autonomous system network for the purpose of introducing IS-IS, iBGP and eBGP to the student. This infrastructure shows the relationship between different autonomous systems in an "Internet". The teams belonging to each network work together as a typical ISP. Each AS has two links to its neighbouring ASes, and this feature will used throughout a significant portion of this workshop.

The connectivity shown in the diagrams represents links between AS's. It is assumed that all the routers within an AS are physically connected to each other as per Figure 1.

Lab Exercises

1. **Topology.** The instructor will have configured the network to the topology shown in Figure 1. All routers within an AS must be physically connected and reachable. The relationship between the ASes is as drawn in Figure 2 and gives a view which can be related to the "real world".

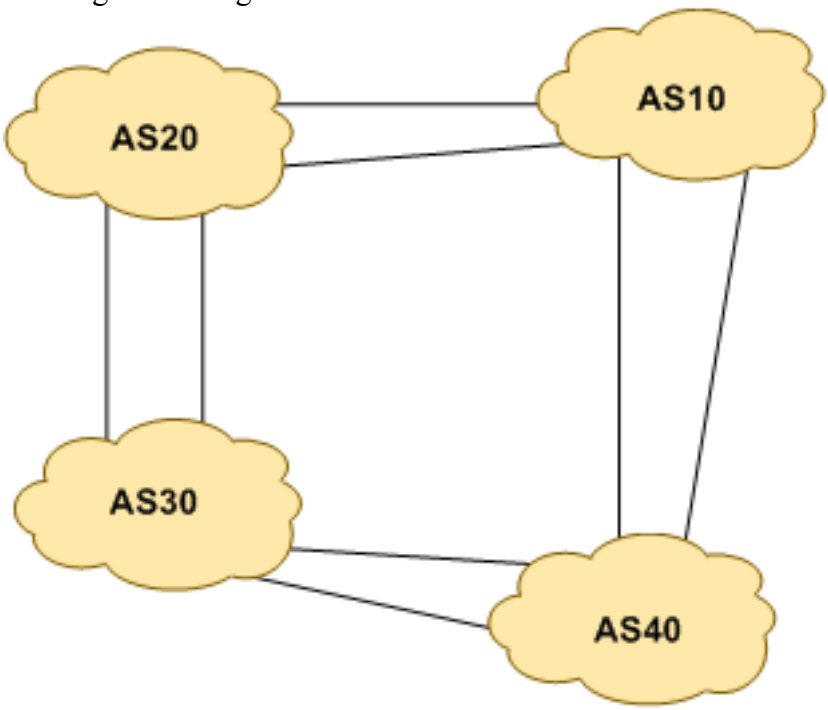

Figure 2 – AS relationship

2. **Routers and the Workshops participants.** This workshop is laid out such that a group of two students will operate a single router. 14 routers generally imply at least 28 participants. For workshops with larger numbers of participants, groups of three should configure a single router. The Workshop Instructors will divide the routers amongst the workshop participants. In the following notes, a "router team" refers to the group assigned to one particular router.

3. **Introducing the lab.** This workshop uses Cisco IOS routers running IOS, but on the Dynamips systems – Dynamips translates the Cisco 7200 router MIPS processor instructions in IOS to those of the host system, allowing Cisco IOS images, and therefore network configurations, to be run on a host PC system (usually Linux or MacOS based).

The lab will have been preconfigured by the instructors, allowing participants to enter the following exercises directly. Please read the following steps carefully.

4. **Accessing the lab.** The instructors will assign routers to each class group, and will indicate the method of access to the Dynamips server. This will usually be by wireless – if this is the case, make a note of the SSID and any password required. Also make a note of the IP address (IPv4, as Dynamips only supports IPv4 access) of the Dynamips server.

   Access to Dynamips will be by telnet, to a high port, which the instructor will specify. Each participant should ensure that their device has a suitable telnet client. Linux and MacOS system have access to a shell command prompt (or Terminal) programme, which allows telnet at the command line. Windows users can use the Windows "Command Prompt" with the telnet client there, but it's notoriously unreliable. Better to install software such as Putty, TeraTerm, HyperTerm or similar third party telnet client.

   Using the client, connect to the router you have been assigned; for example, to connect to the console port of Router 1:

   ```
   telnet 10.10.0.241  2001
   ```

   or to Router 12:

   ```
   telnet 10.10.0.241  2012
   ```

   Once connected, you will see the Dynamips response, followed by the login or command prompt of the router:

   ```
   bash-3.2$ telnet 10.10.0.241 2001
   Trying 10.10.0.241...
   Connected to dynamips.
   Escape character is '^]'.
   Connected to Dynamips VM "r1" (ID 0, type c7200) - Console port


   User Access Verification

   Username:
   ```

   If the "Connected to Dynamips VM" won't appear, even after hitting the Return key several times, please request help from the workshop instructors.

5. **Router Hostname.** Each router will be named according to the table location, Router1, Router2, Router3, etc. Documentation and labs will also refer to *Router1* as R1. At the router prompt, first go into enable mode, then enter "config terminal", or simply "config" by itself:

   ```
   Router> enable
   Router# config terminal
   Enter configuration commands, one per line.  End with CNTL/Z.
   Router(config)# hostname Router1
   Router1(config)#
   ```

6. **Turn Off Domain Name Lookups.** Cisco routers will always try to look up the DNS for any name or address specified in the command line. You can see this when doing a *trace* on a router

with no DNS server or a DNS server with no in-addr.arpa entries for the IP addresses. We will turn this lookup off for the labs for the time being to speed up traceroutes.

```
Router1 (config)# no ip domain-lookup
```

7. **Disable Command-line Name Resolution.** The router by default attempts to use the various transports it supports to resolve the commands entered into the command line during normal and configuration modes. If the commands entered are not part of Cisco IOS, the router will attempt to use its other supported transports to interpret the meaning of the name. For example, if the command entered is an IP address, the router will automatically try to connect to that remote destination. This feature is undesirable on an ISP router as it means that typographical errors can result in connections being attempted to remote systems, or time outs while the router tries to use the DNS to translate the name, and so on.

```
Router1 (config)# line con 0
Router1 (config-line)# transport preferred none
Router1 (config-line)# line vty 0 4
Router1 (config-line)# transport preferred none
```

8. **Disable Source Routing.** Unless you really believe there is a need for it, source routing should be disabled. This option, enabled by default, allows the router to process packets with source routing header options. This feature is a well-known security risk as it allows remote sites to send packets with different source address through the network (this was useful for troubleshooting networks from different locations on the Internet, but in recent years has been widely abused for miscreant activities on the Internet).

```
Router1 (config)# no ip source-route
```

9. **Usernames and Passwords.** All router usernames should be ***isplab*** and all passwords should be ***lab-PW***. We will make the enable secret ***lab-EN***. Please do **not** change the username or password to anything else, or leave the password unconfigured (access to vty ports is not possible if no password is set). It is essential for a smooth operating lab that all participants have access to all routers.

```
Router1 (config)# username isplab secret lab PW
Router1 (config)# enable secret lab EN
Router1 (config)# service password encryption
```

The *service password-encryption* directive tells the router to encrypt all passwords stored in the router's configuration (apart from *enable secret* which is already encrypted).

Note A: There is the temptation to simply have a username of *cisco* and password of *cisco* as a lazy solution to the username/password problem. Under no circumstances must any service provider operator ever use easily guessable passwords as these on their live operational network[1].

Note B: for IOS releases prior to 12.3, the username/secret pair is not available, and operators will have to configure username/password instead. The latter format uses type-7 encryption, whereas the former is the more secure md5 based encryption.

[1] This sentence cannot be emphasized enough. It is quite common for attackers to gain access to networks simply because operators have used familiar or easily guessed passwords.

10. **Enabling login access for other teams.** In order to let other teams telnet into your router in future modules of this workshop, you need to configure a password for all virtual terminal lines.

```
Router1 (config)# aaa new-model
Router1 (config)# aaa authentication login default local
Router1 (config)# aaa authentication enable default enable
```

This series of commands tells the router to look locally for standard user login (the username password pair set earlier), and to the locally configured enable secret for the enable login. By default, login will be enabled on all vtys for other teams to gain access.

11. **Configure system logging.** A vital part of any Internet operational system is to record logs. The router by default will display system logs on the router console. However, this is undesirable for Internet operational routers, as the console is a 9600 baud connection, and can place a high processor interrupt load at the time of busy traffic on the network. However, the router logs can also be recorded into a buffer on the router – this takes no interrupt load and it also enables to operator to check the history of what events happened on the router. In a future module, the lab will configure the router to send the log messages to a SYSLOG server.

```
Router1 (config)# no logging console
Router1 (config)# logging buffer 8192 debug
```

which disables console logs and instead records all logs in a 8192byte buffer set aside on the router. To see the contents of this internal logging buffer at any time, the command "`sh log`" should be used at the command prompt.

12. **Save the Configuration.** With the basic configuration in place, save the configuration. To do this, exit from enable mode by typing "end" or "<ctrl> Z", and at the command prompt enter "write memory".

```
Router1(config)#^Z
Router1# write memory
Building configuration...
[OK]
Router1#
```

It is highly recommended that the configuration is saved quite frequently to NVRAM, especially in the workshop environment where it is possible for power cables to become dislodged. If the configuration is not saved to NVRAM, any changes made to the running configuration will be lost after a power cycle.

Log off the router by typing exit, and then log back in again. Notice how the login sequence has changed, prompting for a "username" and "password" from the user. Note that at each checkpoint in the workshop, you should save the configuration to memory – remember that powering the router off will result in it reverting to the last saved configuration in NVRAM.

13. **IP Addressing.** We need to come up with a sensible and scalable addressing plan for each AS in this network. Each AS gets their own /20 address block (typical minimum allocation for a starter ISP). This address block should be assigned to links and loopbacks on the routers making up each ASN. The allocations are as follows:

AS10	**10.10.0.0/20**	**AS20**	**10.20.0.0/20**

AS30 10.30.0.0/20 **AS40 10.40.0.0/20**

We need to divide up each address block so that we have customer address space, network infrastructure address space, and some space for loopbacks. Figure 3 below reminds how this could be done:

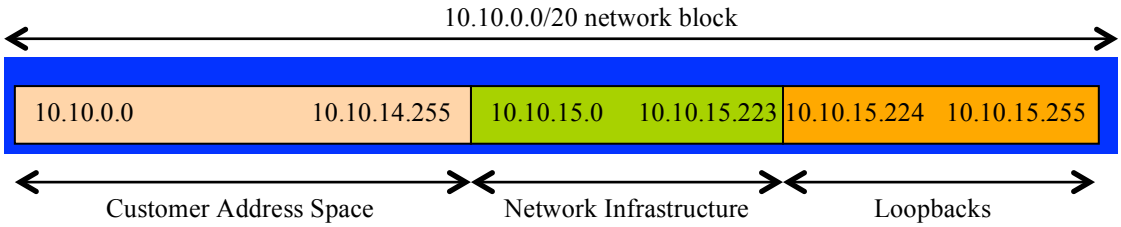

Figure 3 – Dividing allocated block of /20 into Customer, Infrastructure and Loopbacks

Please refer to the accompanying hand out for the address plan which should be used for this module onwards – it is entitled "Addressing Plan – Modules 6 to 9".

14. **Configuring IP addresses on interfaces.** We now configure the addresses on each interface which will be used for this module, and check basic IP connectivity with our immediately adjacent neighbours. Here is an example for a serial interface on Router 2:

```
Router2(config)# interface serial 1/0
Router2(config-if)# ip address 10.0.15.17 255.255.255.252
Router2(config-if)# description 2 Mbps Link to Router4 via DTE/DCE Serial
Router2(config-if)# no ip redirects
Router2(config-if)# no ip directed-broadcast
Router2(config-if)# no ip proxy-arp
Router2(config-if)# no shutdown
```

Q: What network mask should be used on point-to-point links?

A: On serial interfaces, the network mask should be /30 (or 255.255.255.252 in dotted quad format). There is no point in using any other size of mask as there are only two hosts on such a link. A 255.255.255.252 address mask means 4 available host addresses, of which two are usable (the other two representing network and broadcast addresses).

15. **Ethernet Connections.** The Ethernet links between the routers will be made using *cross-over* RJ-45 cables – these will directly connect the Ethernet ports on the two routers without the requirement for an Ethernet switch. IP subnets will again be taken from the Addressing Plan. Don't make the mistake of assigning a /24 mask to the interface address – there are only two hosts on the Ethernet connecting the two routers, so a /30 mask should be entirely sufficient.

16. **Ping Test.** Ping all physically connected subnets of the neighbouring routers. If the physically connected subnets are unreachable, consult with your neighbouring teams as to what might be wrong. Don't ignore the problem – it may not go away. Use the following commands to troubleshoot the connection:

```
show arp                                : Shows the Address resolution protocol
show interface <interface> <number>     : Interface status and configuration
show ip interface                       : Brief summary of IP interface status and configuration
```

17. **Router Loopback Interface Addressing.** We have set aside a /27 for loopbacks even though each AS has either 3 or 4 routers in it – this leaves more than sufficient room for future expansion. The loopback address assignments which will be used for this module are below:

Router1	**10.10.15.224**	**Router8**	**10.30.15.224**
Router2	**10.10.15.225**	**Router9**	**10.30.15.225**
Router3	**10.10.15.226**	**Router10**	**10.30.15.226**
Router4	**10.20.15.224**	**Router11**	**10.40.15.224**
Router5	**10.20.15.225**	**Router12**	**10.40.15.225**
Router6	**10.20.15.226**	**Router13**	**10.40.15.226**
Router7	**10.20.15.227**	**Router14**	**10.40.15.227**

For example, Router Team 1 would assign the following address and mask to the loopback on Router 1:

```
Router1(config)#interface loopback 0
Router1(config-if)#ip address 10.10.15.224 255.255.255.255
```

Q: Why do we use /32 masks for the loopback interface address?

A: There is no physical network attached to the loopback so there can only be one device there. So we only need to assign a /32 mask – it is a waste of address space to use anything else.

~~18. **Configure IS-IS on the routers within each AS.** In each AS configure IS-IS routing. This means that each router team should configure *router ISIS* with IS-IS ID *asY* on the router, where *Y* is the AS number. And the **internal** links to each member of the AS must be configured with *ip router ISIS asY*. The NET should be *49.0001.x.x.x.00*, where *x.x.x* is built from the loopback IP address.~~

> ~~**Hint:** A nice trick for converting the loopback interface address into the NSAP address is to take the loopback address and put the missing leading zeroes in. For example, Router 5 loopback address is 10.10.15.225; this is rewritten to 010.010.015.225 putting in the missing zeroes. Then rather than having the dot after every third character, move it to be after every fourth character. So 010.010.015.225 becomes 0100.1001.5225.~~

~~IS-IS should be configured on internal interfaces **only**. You do not want to set up adjacencies with devices outside your AS. Make sure that there are no *ip router isis* commands on external interfaces. A side effect of this is that external link addresses will not appear in the IGP (see the next section discussion iBGP deployment).~~

~~As an example, Router 1, which has two interfaces inside AS 10, would have the following configuration:~~

```
Router1 (config)# router isis as10
Router1 (config router)# net 49.0001.0100.1001.5224.00
Router1 (config router)# is-type level-2-only
Router1 (config router)# metric-style wide level-2
Router1 (config router)# metric 100000
Router1 (config router)# log-adjacency-changes
Router1 (config router)# set-overload-bit on-startup wait-for-bgp
!
Router1 (config)# interface fastethernet 0/0
Router1 (config if)# ip router isis as10
```

21. Ping Test. Check the routes via IS-IS. Make sure you can see all the networks within your AS, and see no networks from other ASs. Ping all loopback interfaces within your AS Set. Use the "*show clns neighbor"* and *"show ip route"* commands.

22. Telnet source address. Most ISPs use the router Loopback address for administrative purposes as well as the anchor point for their network's iBGP sessions. In this step we will configure telnet so that it uses the loopback interface as the source address for all telnet packets originated by the router.

```
Router1 (config)# ip telnet source-interface loopback 0
```

To check that this has worked, telnet from your router to a neighbouring router and then enter the "who" command. You will see that you are logged in, and the source address will be displayed. For example, using telnet from Router1 to Router3 gives:

```
Router3>who
    Line      User       Host(s)                Idle Location
*   2 vty 0   isplab     idle                00:00:00 10.10.15.224
```

23. Save the configuration. Don't forget to save the configuration to NVRAM!

Checkpoint #1 : *call the lab assistant to verify the connectivity.*

~~**24. Turning on neighbour authentication for IS-IS – Part 1.** IS-IS supports neighbour authentication; this is considered more and more important inside ISP networks as attacks on infrastructure increase and ISPs seek to use all available tools to secure their networks. (While an attack on IS-IS is harder as it runs on the link layer alongside IP rather than on top of IP like OSPF, some ISPs are still prudent and implement neighbour authentication.)~~

~~Each router team will now turn on neighbour authentication for IS-IS. The first step is to set up the keychain to be used – we will use the key "labnetpw" for this lab:~~

```
Router1(config)# key chain lab-key
Router1(config-keychain)# key 1
Router1(config-keychain-key)# key-string labnetpw
```

~~**25. Turning on neighbour authentication for IS-IS – Part 2.** Now that the keychain has been defined, we activate authentication within the IS-IS processes. The first step is to enable MD5 for level-2 IS's:~~

```
Router1(config)# router isis as10
Router1(config-router)# authentication mode md5 level-2
```

~~And then associate the key-chain we defined earlier with the configured authentication:~~

```
Router1(config-router)# authentication key-chain lab-key level-2
```

~~Notice now that the IS-IS adjacencies do not come up unless the neighbouring router has also entered the same configuration and key. Notice also how the IS-IS adjacencies were reset as the configuration was entered – security is being introduced, so the adjacencies are reset.~~

26. Final check. Use the various "*show isis*" commands to see the IS-IS status of the lab network now. Check the routing and the routing table. Make sure all the adjacencies have come back up again. If any adjacency has failed to come up, and you see several log messages saying:

```
*Dec  4 13:01:01.932: %CLNS-4-AUTH_FAIL: ISIS: LAN IIH authentication failed
*Dec  4 13:03:19.955: %CLNS-4-AUTH_FAIL: ISIS: LSP authentication failed
*Dec  4 13:03:50.111: %CLNS-4-AUTH_FAIL: ISIS: PSNP authentication failed
```

you should reasonably expect that either you or your connected neighbour have forgotten to set up neighbour authentication.

Note: Wherever an IS-IS session is configured from now on in the workshop, all Router Teams MUST use passwords on these IS-IS sessions.

Checkpoint #2 : *call the lab assistant to verify the connectivity.*

27. Configure iBGP peering between routers within an AS. Use the loopback address for the iBGP peerings. Also, configure the *network* command to add the address block assigned to each Router Team for advertisement in BGP.

```
Router1 (config)# router bgp 10
Router1 (config-router)# distance bgp 200 200 200
Router1 (config-router)# no synchronization
Router1 (config-router)# network 10.10.0.0 mask 255.255.240.0
Router1 (config-router)# neighbor 10.10.15.225 remote-as 10
Router1 (config-router)# neighbor 10.10.15.225 update-source loopback 0
Router1 (config-router)# neighbor 10.10.15.225 next-hop-self
Router1 (config-router)# neighbor 10.10.15.225 description iBGP Link to R2
Router1 (config-router)# neighbor 10.10.15.226 remote-as 10
Router1 (config-router)# neighbor 10.10.15.226 update-source loopback 0
Router1 (config-router)# neighbor 10.10.15.226 next-hop-self
Router1 (config-router)# neighbor 10.10.15.226 description iBGP Link to R3
Router1 (config-router)# no auto-summary
Router1 (config-router)# exit
Router1 (config)# ip route 10.10.0.0 255.255.240.0 Null0
```

28. Next-hop-self configuration. The previous step introduced the next-hop-self configuration command. Referring to the BGP presentation, the next-hop-self configuration makes the iBGP speaking router use the iBGP source address (in this case the loopback) rather than the external next-hop address (as per the BGP specification). This is industry best practice and means that ISPs do not need to carry external next-hops in their IGP.

29. Test internal BGP connectivity. Use the BGP Show commands to ensure you are receiving everyone's routes from within your AS.

30. Configure Deterministic MED. Another industry best practice is to configure deterministic MED for BGP. This means that IOS will order by AS Number the same prefix heard from multiple paths, and do the best path selection per ASN group. The IOS default is to compare the paths for the same prefix from most recent to the oldest, which can result in non-deterministic (ie different) path selection each time the path selection process is run. For example, for Router5:

```
Router5 (config)# router bgp 20
```

```
Router5 (config-router)# bgp deterministic-med
```

Note that it is unlikely that deterministic MED will have any impact on the path selection for this Module. However, it is an industry best practice now, and network operators should include it in their BGP configuration template by default.

31. **Configure passwords on the iBGP sessions.** Passwords should now be configured on the iBGP sessions. Review the presentation why this is necessary. Agree amongst all your team members in your AS what the password should be on the iBGP session, and then apply it to all the iBGP peerings on your router. For example, on Router2's peering with Router3, with "bgp-pw" used as the password:

```
Router2 (config)# router bgp 10
Router2 (config-router)# neighbor 10.10.15.226 password bgp-pw
```

IOS currently resets the iBGP session between you and your neighbouring router whenever an MD5 password is added. So when passwords are added to BGP sessions on live operational networks, this work should be done during a maintenance period when customers know to expect disruptions to service. In the workshop lab, it doesn't matter so much. (Future IOS releases will avoid having this rather serious service disruption.)

Watch the router logs – with the BGP session neighbour changes being logged, any mismatch in the password should be easy to spot. A missing password on one side of the BGP session will result in the neighbouring router producing these errors:

```
%TCP-6-BADAUTH: No MD5 digest from 3.3.3.3:179 to 2.2.2.2:11272
%TCP-6-BADAUTH: No MD5 digest from 3.3.3.3:179 to 2.2.2.2:11272
%TCP-6-BADAUTH: No MD5 digest from 3.3.3.3:179 to 2.2.2.2:11272
```

whereas a mismatch in the configured passwords will result in these messages:

```
%TCP-6-BADAUTH: Invalid MD5 digest from 3.3.3.3:11024 to 2.2.2.2:179
%TCP-6-BADAUTH: Invalid MD5 digest from 3.3.3.3:11024 to 2.2.2.2:179
%TCP-6-BADAUTH: Invalid MD5 digest from 3.3.3.3:11024 to 2.2.2.2:179
```

Checkpoint #3: *Call the lab assistant and demonstrate the password as set on the iBGP session. Once confirmed by the lab assistant, move on to the next steps.*

32. **Configure eBGP peering.** Use Figure 1 to determine the links between the AS's. Addressing for eBGP links between 2 AS's will use the point-to-point interface addresses, ***NOT*** the loopback addresses (review the BGP presentation if you don't understand why).

```
Router1 (config)# router bgp 10
Router1 (config-router)# neighbor 10.10.15.14 remote-as 40
Router1 (config-router)# neighbor 10.10.15.14 description eBGP to Router13
```

Use the BGP Show commands to ensure you are sending and receiving the BGP advertisements from your eBGP neighbours.

Q. Why can't the loopback interfaces be used for the eBGP peerings?

A. The IP address of a router's loopback interface is not known to external BGP peers, so the external peers will have no way of knowing how to contact each other to establish the peering.

Q. Which BGP show command allows you to see the state of the BGP connection to your peer?

A. Try *show ip bgp neighbor x.x.x.x* – this will give detailed information about the state of the peer. There are subcommands of this one, giving more information about the peering.

Q. Which BGP Show command will allow you to see exactly which networks you are advertising and receiving from your eBGP peers?

A. Try *show ip bgp neighbor x.x.x.x route* – this will show which routes you are receiving from your peer. Likewise, replacing *route* with *advertised-routes* will list the networks which are being announced to your peer. (Note that in general ISP operational practice, there are caveats here – if you apply route-maps and some BGP policies, these will not be processed by the *advertised-routes* command. Use the *advertised-routes* subcommand with due caution.)

33. **Configure passwords on the eBGP session.** Passwords should now be configured on the eBGP sessions between your and your neighbouring ASes. Agree between you and your neighbouring AS what the password should be on the eBGP session, and then apply it to the eBGP peering. For example, on Router2's peering with Router4, with "bgp-pw" used as the password:

```
Router2 (config)# router bgp 10
Router2 (config-router)# neighbor 10.10.15.10 password bgp-pw
```

As previously for the iBGP session, watch the logs for password mismatches, or missing passwords. As with the iBGP sessions previously, you will find that the router will reset the eBGP session as soon as the password is applied.

Note: Wherever a BGP (either iBGP or eBGP) session is configured from now on in the workshop, all Router Teams MUST use passwords on these BGP sessions.

Checkpoint #4: *Call the lab assistant and demonstrate the password as set on the eBGP session. Once confirmed by the lab assistant, move on to the next steps.*

34. **Adding a "customer" route into BGP.** As we did in Module 1, we are now going to add a "customer" route into BGP on each router. We don't have any "customers" as such connected to our routers in the lab, so we are going to simulate the connectivity by simply using a Null0 interface. The "customer" address space that each router team will introduce into the iBGP is listed below – again we will each use a /26, for simplicity's sake.

R1	**10.10.0.0/26**	**R8**	**10.30.0.0/26**
R2	**10.10.0.64/26**	**R9**	**10.30.0.64/26**
R3	**10.10.0.128/26**	**R10**	**10.30.0.128/26**
R4	**10.20.0.0/26**	**R11**	**10.40.0.0/26**
R5	**10.20.0.64/26**	**R12**	**10.40.0.64/26**
R6	**10.20.0.128/26**	**R13**	**10.40.0.128/26**
R7	**10.20.0.192/26**	**R14**	**10.40.0.192/26**

Each team should now set up a static route pointing to the **NULL0** interface for the /26 that they are to originate. Once the static is set up, the team should then add an entry into the BGP table. Here is an example for Router8:

```
Router8 (config)# ip route 10.30.0.0 255.255.255.192 Null0
Router8 (config)# router bgp 30
Router8 (config-router)# network 10.30.0.0 mask 255.255.255.192
```

35. **Check the BGP table.** Are there routes seen via ***show ip bgp***? If not, why not? Once every team in the class has done their configuration, each team should see the aggregate from each AS as well as the fourteen /26s introduced in the previous step. If this is not happening, work with your neighbours to fix the problem.

Checkpoint #5: *Call the lab assistant to verify the connectivity. Use commands such as "show ip route sum", "show ip bgp sum", "show ip bgp", "show ip route", and "show ip bgp neigh x.x.x.x route | advertise". There should be 4 aggregate prefixes (one for each ISP) and the 14 customer /26's in the BGP table.*

36. **BGP Update Activity (Optional).** Use *debug ip bgp update* to see BGP update activity after clearing a BGP session. To stop the debug running, do *undebug ip bgp update.*

    Warning: it might not be such a good idea to run this debug command on a router receiving the full Internet routing table; using this command in a lab network such as this might show you why!